

CONTREX: Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties

Original

CONTREX: Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties / Goren, Ralph; Gruttner, Kim; Herrera, Fernando; Penil, Pablo; Medina, Julio; Villar, Eugenio; Palermo, Gianluca; Fornaciari, William; Brandolese, Carlo; Gadioli, Davide; Bocchio, Sara; Ceva, Luca; Azzoni, Paolo; Poncino, Massimo; Vinco, Sara; Macii, Enrico; Cusenza, Salvatore; Favaro, John; Valencia, Raul; Sander, Ingo; Rosvall, Kathrin; Quaglia, Davide. - ELETTRONICO. - (2016), pp. 286-293. (Intervento presentato al convegno Conference on Digital System Design (DSD), 2016 tenutosi a Cyprus nel 31 Agosto - 2 Settembre 2016) [10.1109/DSD.2016.95].

Availability:

This version is available at: 11583/2651278 since: 2020-02-22T21:39:29Z

Publisher:

IEEE/Euromicro

Published

DOI:10.1109/DSD.2016.95

Terms of use:

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

CONTREX: Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties

Ralph Görden*, Kim Grüttner*, Fernando Herrera†, Pablo Peñil†, Julio Medina†, Eugenio Villar†,
Gianluca Palermo‡, William Fornaciari‡, Carlo Brandolese‡, Davide Gadioli‡,
Sara Bocchio§, Luca Ceva¶, Paolo Azzoni||, Massimo Poncino**, Sara Vinco**, Enrico Macii**,
Salvatore Cusenza††, John Favaro††, Raúl Valencia‡‡, Ingo Sander^x, Kathrin Rosvall^x, Davide Quaglia^{xi}
* OFFIS – Institute for Information Technology, Oldenburg, Germany; † University of Cantabria, Santander, Spain
‡ Politecnico di Milano, Italy; § STMicroelectronics, Italy; ¶ Vodafone Automotive Telematics, Switzerland
|| Eurotech, Italy; ** Politecnico di Torino, Italy; †† Intecs, Italy; ‡‡ GMV, Spain
^x KTH Royal Institute of Technology, Stockholm, Sweden; ^{xi} EDALab s.r.l., Italy

Abstract—The increasing processing power of today’s HW/SW platforms leads to the integration of more and more functions in a single device. Additional design challenges arise when these functions share computing resources and belong to different criticality levels. The paper presents the CONTREX European project and its preliminary results. CONTREX complements current activities in the area of predictable computing platforms and segregation mechanisms with techniques to consider the extra-functional properties, i.e., timing constraints, power, and temperature. CONTREX enables energy efficient and cost aware design through analysis and optimization of these properties with regard to application demands at different criticality levels.

I. INTRODUCTION

Up to now, mission & safety critical services of electronic systems have been running on dedicated and often custom designed HW/SW platforms. In the near future, such systems will be accessible, connected with or executed on devices comprising off-the-shelf HW/SW components to reduce development costs. A basic requirement for this is the absence of interference among applications of different criticalities sharing computing resources. Significant improvements have been achieved supporting the design of mixed-critical systems by developing predictable computing platforms and mechanisms for segregation. Such platforms enable techniques for the compositional certification of applications’ correctness, run-time properties and reliability.

CONTREX European project complements these important activities with an analysis and segregation along specific extra-functional properties: real-time, power, and temperature. These properties will be a major cost roadblock when

- 1) scaling up the number of applications per platform and the number of cores per chip,
- 2) running devices battery powered, or
- 3) switching to technology nodes with smaller feature size.

CONTREX enables energy efficient and cost aware design through analysis and optimization of real-time, power, and temperature with regard to application demands at different criticality levels. To reinforce European leadership and

industrial competitiveness the CONTREX approach is integrated into existing model-based design methods that can be customized for different application domains and target platforms. CONTREX focuses on requirements derived from the automotive, aeronautics, and telecommunication domains, evaluates their effectiveness, and drives integration into existing standards for design and certification based on three industrial demonstrators. Valuable feedback to the industrial design practice, standards, and certification procedures is pursued.

Our economic goal is to improve energy efficiency and to reduce cost per system due to a more efficient use of the computing platform.

The CONTREX consortium consists of fifteen partners from six countries. There are six academic institutions, six industrial tool or technology providers, and three industrial demonstrator application providers. The project started in October 2013 and ends in September 2016 [1].

The paper is organized as follows: Section II gives an overview of the CONTREX methodology and the demonstrator applications. Section III, Section IV, and Section V give more details about specification and modeling of extra-functional properties and criticalities, simulation and analysis of that properties, and their consideration at runtime. Furthermore, each of the sections describes the usage of the methodological elements in one of the demonstrator applications. Section VI closes the paper with a conclusion and summary.

II. CONTREX METHODOLOGY OVERVIEW

Fig. 1 shows an overview of the CONTREX methodology for the design of mixed critical systems under consideration of extra-functional properties (EFP). Some elements of the methodology have been available before the project started, for instance inputs for the methodology like system models from previous and existing hardware or software components shown in the upper part of the figure. In addition, there are various hardware platforms on-hand, e.g., the Xilinx Zynq platform or the iNemo platform provided by ST, as well as techniques to

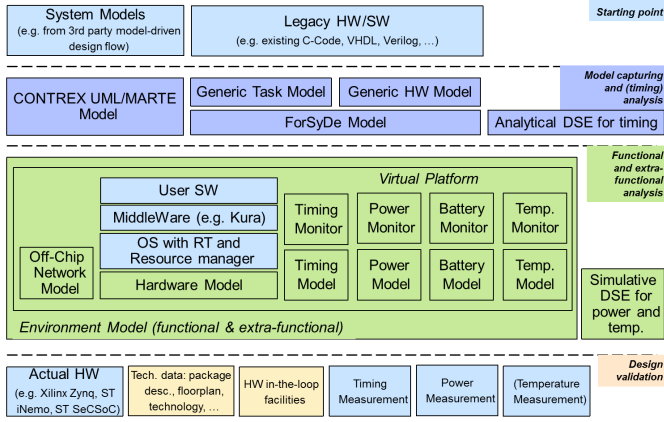


Fig. 1. CONTREX Reference Architecture.

measure the timing, power, and temperature behavior of physically available devices. In between, we have user software, middle-ware components such as the Kura framework, and operating systems with runtime and resource management. The CONTREX project complements this methodology in three aspects: model capturing and timing analysis, functional and extra-functional analysis, and design validation.

For the model capturing, existing meta-models are extended to support the specification of criticalities as well as extra-functional properties. The integration of these models into the ForSyDe framework allows analytical design space exploration for timing. More details about the modeling methodology will be given in Section III using the avionics demonstrator as an example. The functional and extra-functional analysis part is extended to enable simulative design space exploration under consideration of power and temperature properties. To do so, virtual platforms are set up with hardware and communication models and enriched with models for timing, power, batteries, and temperature as well as infrastructure for the runtime-monitoring of these properties. These models can be connected to environment models to simulate the entire system. To complete the flow, technical data of the platforms such as IC package descriptions, floorplans, or technology information, as well as hardware-in-the-loop facilities are added to perform more detailed design validation. In Section IV, the virtual platform based simulation and analysis is described by using the telecom demonstrator. Section V focuses on the techniques for monitoring and management of extra-functional properties at runtime and their application to the automotive telematics demonstrator.

A. Demonstrator Applications

The evaluation of the proposed methodology is based on its adoption in three demonstrator applications: a Remotely Piloted Aircraft, a telecom system (Ethernet-over-Radio), and an automotive telematics system.

1) *Avionics Demonstrator*: The avionic demonstrator concerns a subset of the Flight Control Computer (FCC) software developed for a medium sized Remotely Piloted Aircraft (RPA) applicable for surveillance missions such as damage

assessment and intelligence. Expected improvements for extra-functional budget analysis will result in reduced weight, power, size, and heat dissipation.

2) *Telecom Demonstrator*: The Telecom demonstrator is based on the Ethernet Over Radio System. It is specifically designed and engineered to allow a smooth transition from old digital protocols to new wireless standards up to WDCMA and LTE. It consists of Indoor and Outdoor Units, connected by a Gigabit Ethernet cable which delivers also power. It naturally represents a mixed critical system. Timing guarantees under power and temperature constraints of the hosting equipment, as well as installation weight and space footprint are crucial. The new CONTREX techniques for global optimization over the entire installation greatly enhance cost/performance characteristics.

3) *Automotive Telematics Demonstrator*: This demonstrator provides private and/or fleet vehicle drivers with a support service in case of accident. The architecture is based on an end-to-end cloud-based IoT solution that is responsible for data collection, data processing, and automotive services provisioning. In the vehicle the system relies on three main components: a sensing unit for acceleration measurements, a GPS based localization unit, and a data processing and communication unit for identification of accidents and communication of position data to either public authorities (hospital, police) or private support providers. CONTREX results help to improve performance, energy efficiency, and cost of the system.

III. MODELING OF EFPS AND CRITICALITIES

In the current avionics development flow, both the HW/SW partitioning decision and the platform configuration are made at an early stage of the cycle, mostly based on designers' expertise. Then, HW and SW developments evolve in parallel until the integration phase. In order to avoid late integration issues, the design space is strongly limited to a very small number of possibilities. In addition to this, the quantity and capacity of resources (such as buses, number of processors or memory space) are usually oversized, particularly in the case of mixed-criticality systems, due to the spatial and temporal isolation principle. This development flow is especially geared to developing custom platforms for the systems under construction, and presents several difficulties to its applicability to new contexts, such as the development of light Remotely Piloted Aircraft (RPA) equipment. Taking into account that size, weight and power (SWaP) constraints are a key factor for RPA equipment, its Flight Control Computer (FCC) software is susceptible to being reused in diverse commercial all-purpose HW platforms and low-cost avionics sensors to be integrated in light RPAs for new markets and countries, enabling to maintain a competitive supply base in Europe. To support these new required capabilities, it is expected that CONTREX modeling methodology improves the current avionics development flow by introducing extra stages for system modeling, model-based analysis, simulation and Design Space Exploration (DSE) during the design phase, as shown in Fig. 2.

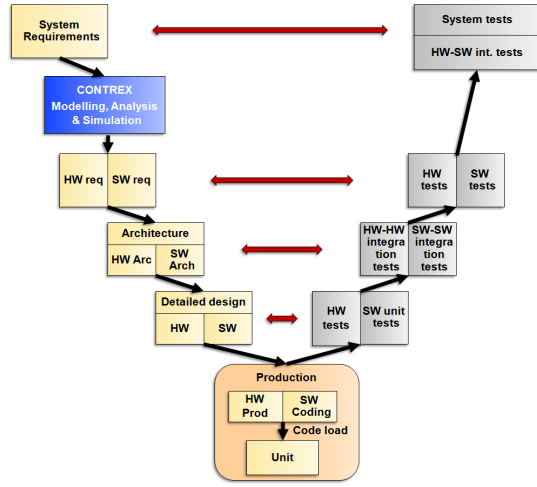


Fig. 2. Integration of the CONTREX modeling, analysis, simulation and DSE methods and tools in the avionics industrial flow.

Specifically, CONTREX is providing a design approach covering the aforementioned modeling, analysis, simulation and DSE activities. The main language used in CONTREX for the modeling activity is UML, complemented with the MARTE profile. Moreover, as sketched in Fig. 3, CONTREX connects the UML/MARTE and ForSyDe [2] methodologies, and also relies on CAMEL-View [3] for the capture of complex physical environments, to exercise the simulation-based performance model.

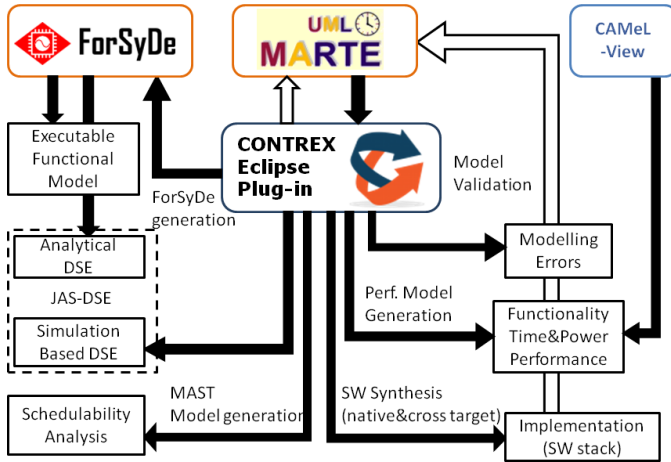


Fig. 3. Languages and design activities in the single-source design approach developed in CONTREX

The CONTREX UML/MARTE model captures all the information required for tackling a number of system-level design activities. From these models, those design activities can be performed with the modeling and design framework developed in CONTREX, called in short CONTREP (CONTREX Eclipse plug-in) [4], and shown in Fig. 3. A former and crucial design activity is a proper capture of the model. CONTREP supports the modeling activity with a model validation facility. The model can be used for software synthesis [5]. However, finding a suitable and efficient implementation is required first.

CONTREP enables an automated generation of a simulatable performance model [6] relying on the VIPPE tool [7][8]. VIPPE relies on *native* (or *source-level*) *performance simulation*, a performance estimation technology capable to offer performance estimations close in accuracy, but one or more order of degrees faster, than instruction-set simulators (ISS) or simulators relying on *binary translation*. This makes native simulation convenient for design space exploration with concern on EFPs. CONTREP also enables the automated generation of an automated DSE framework, in turn relying on the automatically generated VIPPE model [9]. CONTREP also connects with a schedulability analysis tool [10].

Among the relevant aspects to tackle the challenges mentioned at the beginning of the section, here we highlight (1) the capability for capturing mixed-criticality in the modeling methodology; and (2) the capability of the developed methods and tools to exploit such information with concern on extra-functional properties and the requirements on them.

CONTREX proposes a minor extension of the MARTE standard allowing the notation of criticality as a generic attribute which allows the adaptation to different modeling scenarios [11]. This way, the CONTREX UML/MARTE methodology allows to capture criticality and associate it to different model elements. The methodology also supports the annotation of worst-case execution times per criticality, as it is required by recent mixed-criticality schedulability analysis algorithms. Moreover, CONTREX covers a scenario where mixed-criticality refers to the possibility to associate criticalities to performance requirements. For instance, Fig. 4 shows the modeling of timing requirements associated to particular tasks of the RPA IO system (CONTREX avionics demonstrator) and a global power requirement for it too. These time and power performance requirements have associated different criticalities.

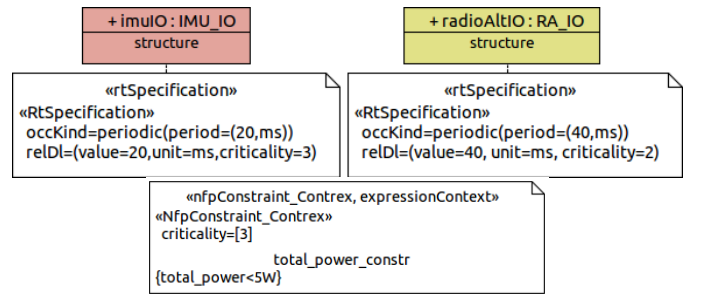


Fig. 4. CONTREX UML/MARTE models enable the association of different criticalities to performance requirements.

CONTREX UML/MARTE modeling methodology also supports MARTE-based specification of a design space for supporting an efficient DSE [9]. A holistic DSE is enabled, because the design space comprises extra-functional parameters which can refer to different levels of the system. For instance, for the RPA_IO system, parameters on the application (task periods), and parameters on the platform, i.e. processor working frequencies (shown in Fig. 5), are explored.

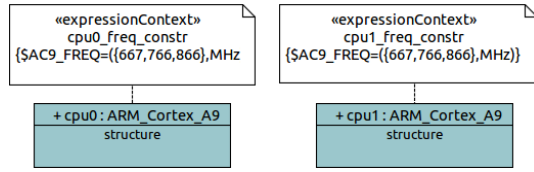


Fig. 5. Parameterization of processor frequencies for DSE.

As well as enabling modeling of mixed-criticality systems, and modeling of DSE parameters (and thus modeling of a design space) for efficient DSE, CONTREX tools are able to exploit mixed-criticality information at several design activities, DSE being one among them.

The design framework is capable to exploit this information at model validation, to check that mixed-criticality aware modeling rules are fulfilled. The framework enables the association of criticality levels to different severity levels. This way, the severity level in the report of a performance constraint violation, that has been detected in the simulation of the automatically produced performance model, depends on the criticality associated to that constraint [12].

As sketched in Fig. 3, the UML/MARTE methodology has been connected with the ForSyDe methodology [2]. CONTREX metamodel has enabled the definition of a common base [9], relying on the theory of models of computation (MoCs) [13], which enables to convert MARTE models into ForSyDe-SystemC models [14]. MoC theory ensures relevant functional properties and analyzability for more critical parts of the system functionality. Then, these parts can be converted into ForSyDe-SystemC and simulated.

Moreover, ForSyDe-SystemC models have been enabled as a design entry to a static and analytical design space exploration of mixed-criticality systems [15], which can be used as ultra fast DSE.

The analytical DSE can be also used for a preliminary design space prune in a joint-analytical and simulation-based DSE (JAS-DSE). CONTREX contributes a JAS-DSE flow, which combines the aforementioned analytical DSE with simulation-based DSE, relying on the tools KisTA [16] and MOST [17]. Moreover, mixed-criticality simulation-based DSE is also enabled by employing mixed-criticality in the configuration of the cost functions. The simulatable model is automatically generated. The model through automatic generation of fast performance assessment models, which enable an early, fast, but accurate, design space exploration.

The application of the CONTREX approach to the avionics development flow is expected to provide a more flexible framework that enables an early assessment of the system performance for the different candidate platforms taken into consideration, as well as an efficient exploration of wider design spaces, with the purpose of finding optimal configurations that minimize cost, size, weight and power consumption of the system without compromising its safety or overall performance. In the particular case of the mixed-criticality systems, the possibility of assessing system performance in multi-

core architectures (including commercial general-purpose platforms) in an early development phase might lead to significant time savings and cost reductions. The results gathered during the enhanced design phase would enable the designer to make informed architectural decisions (such as platform selection and HW-SW mapping) based on reliable performance figures, so design errors are minimized, and the chance for design rework at late stages is significantly reduced.

IV. EXTRA-FUNCTIONAL PROPERTY MODELING, SIMULATION, AND MONITORING

Several challenging extra-functional property requirements apply to modern telecom systems. The assurance of thermal and power properties is challenging from the environmental point of view, because components are often characterized by outdoor placement. The electronic circuits present challenges because the clock rate is heavily influenced by the technology used for CPU, busses, and FPGA components. Likewise, the assurance of timing properties is difficult because of extreme variation in traffic loads that render analysis complex or even intractable. These challenges can lead to costly errors in the dimensioning of systems that are only discovered after deployment.

A virtual prototyping environment on the host processor makes it possible to test the functional and extra-functional behavior of the system under development. The real hardware prototype can be delayed to later design phases, permitting early, low cost evaluation of the system's timing behavior. Using estimation tools operating in the simulated environment, it also permits exploration of different hardware architecture configurations that optimize thermal and power characteristics.

A. Virtual Platforms

The Virtual platform (VP) is an executable model of a system that can be used for early software development and architectural analysis. Every VP includes processor, bus, memory and peripheral models, potentially supporting pre-silicon development of the entire software stack up to the applications level. VPs usually provide a debugging environment to improve software quality and reduce software development costs and time to market.

Imperas Open Virtual Platform (OVP) [18] consists of a set of open source C-based platform descriptions and a closed-source CPU emulator supporting the fast instruction-accurate simulation of several CPU architectures. CPU emulation is based on dynamic binary translation [19]. The CPU emulator and peripheral models are also available as modules written by using Accellera Transaction-Level Modeling (TLM) standard [20].

While OVP and all commercial VPs provide models for standard components, the integration of custom IP blocks is an issue because they are usually described at RTL in VHDL or Verilog while VP models are usually written in SystemC or C/C++ at TLM level. Commercial VPs provide co-simulation mechanisms to handle different languages at the cost of slower simulation while hand-made transactors to connect RTL blocks

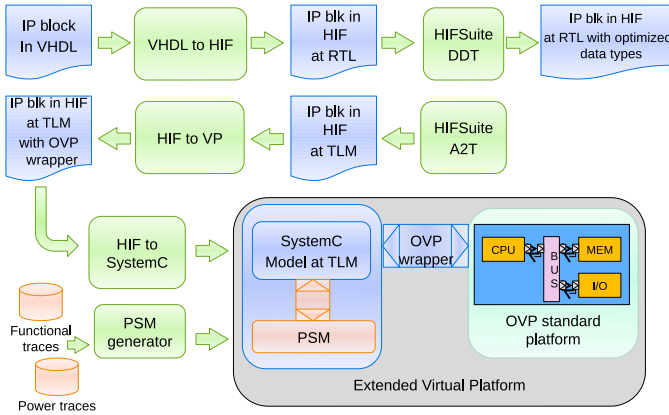


Fig. 6. VP generation flow.

are inefficient and error-prone. Furthermore, standard VPs do not provide models reproducing the behavior of extra-functional properties (e.g., power and temperature) together with functional behavior. Currently, power and temperature are simulated off-line by using ad-hoc tools without exploring the interaction with embedded software.

B. Automatic VP integration supporting EFP

The previously reported issues are solved by extending EDALab's model manipulation tool named HIFSuite to generate extended VPs. The overall flow is reported in Fig. 6. HIFSuite allows to import models described in VHDL or Verilog into a XML-based representation named Heterogeneous Intermediate Format (HIF). Then the tool allows to manipulate such representation to abstract it at TLM level (DDT [21] and A2T [22]) and to generate the OVP wrapper to connect the module to a standard OVP platform.

The CONTREX project introduced a methodology for the automatic generation of Power State Machines (PSMs) [23] by adopting an approach based on (i) dynamic mining of temporal assertions to extract the IP's functional behaviors from a set of functional traces, and (ii) a calibration process to extract the associated power behaviors from a corresponding set of references power traces. Finally, a Markov model was defined to implement a SystemC executable model of the PSMs to be integrated in a standard VP like a traditional functional description. The power estimation obtained by a system-level simulation of the automatically generated PSMs is up to two orders of magnitude faster than running a state-of-the-art gate-level power simulator like PrimeTime PX without a significant loss of accuracy. This approach enables the efficient simulation of power behavior together with functional behavior to find interferences between applications at different criticality levels and to test adaptation policies made at software level.

C. Stream-based Simulation and Tracing Framework

To enable the seamless integration of extra-functional property models into virtual platforms, a framework for stream-based simulation and tracing has been developed [24]. It allows the instrumentation of virtual platforms to access functional

and extra-functional aspects at simulation runtime, as well as pre-processing, monitoring, and recording of these properties.

The underlying technique is based on *timed value streams*, i.e., a sequence of (value, duration) tuples. A *stream writer* is a source of such a stream, a *stream reader* a sink. A *stream processor* is both, sink of one or more streams and source of one or more streams. The basic idea of using the framework is that leaf annotations in the functional model push tuples according to the current local simulation time and status or activity of the producing process to a stream. These incoming tuples are buffered within the stream without advancing the stream's local time. Once the stream writer explicitly commits its updates, the stream's local time is advanced and the pending tuples are forwarded to the stream readers. Stream processors can be used for online pre-processing, filtering, or temporal or structural abstraction. Then, stream sinks can be used for online monitoring or generation of trace files.

The main advantages of this framework over others such as `sc_trace` are:

- high flexibility due to composability of streams and dynamic adaptivity of parameters at runtime,
- support for physical quantities, e.g. with Boost.Units, and
- distributed time model to support temporal decoupling.

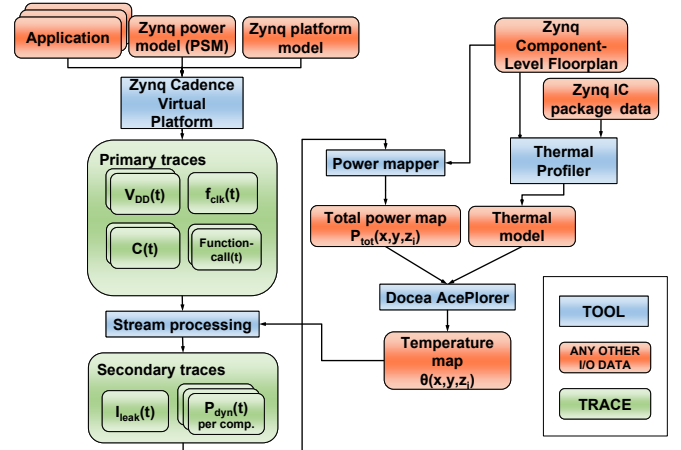


Fig. 7. Stream processing flow.

Fig. 7 shows the application of this concept within the CONTREX project. The application is running on an OVP-based virtual multi-processor platform representing the Xilinx Zynq SoC. The OVP API gives access to the basic parameters of the platform activity, such as each processor's workload or the number memory accesses in a certain period of time. These basic parameters are fed to a set of *primary streams*. A stream processor reads these primary streams, and, together with some further parameters like supply voltage and processor frequency, it calculates the power dissipation per processor. Then, the power values are written to *secondary streams*. Finally, the secondary streams are connected to a VCD sink that outputs the power over time traces as VCD file.

With this, the result of the virtual platform simulation is not only the functional behavior but as well a component

level power trace for an application running on that platform. In addition to the power analysis, we can locate the power dissipation to corresponding areas in the SoC's floorplan, and, together with a thermal model off the SoC package, we can simulate the thermal behavior of the chip as well.

D. Application to Telecom Demonstrator

The Telecom demonstrator is a Point-to-Point Ethernet over Radio Microwave Wireless System. Within the CONTREX project, the objective is the exploration of the tradeoffs of moving from a legacy PowerPC CPU to a modern Xilinx Zynq architecture with higher performance but also higher power consumption and thermal dissipation. To this end, the application is being simulated in the OVP environment described above. Since the system contains adaptive algorithms that regulate the transmission according to the importance of transmitted data and to channel condition, it needs to be simulated in a full and realistic network scenario to test the behavior of transmission control tasks as a function of time-varying transmission condition. The network interface is a legacy HW component written in VHDL, which is being abstracted and translated into SystemC/C++ to be integrated into the OVP scenario by using HIFSuite. The OVP top-level configuration is automatically generated by HIFSuite. The bridging elements to allow co-simulation between OVP and SystemC/C++ components are automatically generated by HIFSuite. The power and thermal analysis is used for comparative estimates of power consumption and thermal behavior of the demonstrator application in different configurations on the Zynq architecture, both in the simulated and the real environments.

V. RUNTIME AND SERVICE ABSTRACTION

The Automotive Demonstrator has been defined to stress the run-time part of the CONTREX methodologies. In addition to that, it includes also the concept of a distributed system where sensor nodes and remote control units deployed on the cars have to communicate with remote infrastructure for data collection. Thus, this use case not only considers node-level extra-functional modeling, monitoring and management but also the remote services abstractions.

A. Cloud Service Abstraction

The concept of connected devices is changing the embedded systems world. Machine to Machine (M2M) and Internet of Things (IoT) follow a common technological paradigm: intelligent devices, seamlessly connected to the Internet, enable remote services and provide actionable data. The IoT acronym is more adopted in the consumer space while M2M has a stronger industrial connotation, such as for the Automotive scenario. One of the most important aspects of the IoT/M2M vision is that smart objects communicate effectively with each other and possibly with applications residing in data centers or the cloud. This however creates a need of a standardized software layer involving both the *Device-to-Cloud* related part and the *Cloud Platform*.

The concept of the *device to cloud* proposes an end-to-end solution that includes purpose-built hardware, connectivity and embedded device management through a pervasive software framework and a cloud client, running on the devices, and a set of machine to machine (M2M) cloud-based services. The objective of this solution is to deliver actionable data from the field to downstream applications and business processes, dashboards and reports. The Kura pervasive framework [25] proposed in CONTREX offers the technical building blocks required to assemble distributed systems of devices and sensors which are effectively connected to IT infrastructures. This solution is based on a combination of hardware, firmware, operating systems, programming frameworks that dramatically accelerate the time to market of M2M / IoT projects and enable future potential customers to layer their added-value components on a reliable ready-to-use infrastructure.

The *cloud platform* is a M2M integration platform that simplifies device and data management by connecting distributed devices over secure and reliable cloud services. The devices can be IoT modules deployed in the environment, e.g. the embedded systems installed in the car. The data are the functional and extra functional properties monitored by these devices. The cloud service abstraction is responsible to provide full control over the embedded systems hardware, software and acquired data with a simple service model. The objective is to completely hide the complex details that stand behind the remote management procedures, remote data acquisition and transmission.

B. Extra-functional properties management at run-time

Power management at node level is of particular interest in battery powered sensor-node. In detail, the main idea within the sensor-node of the automotive use case has been to configure the node's hardware devices and software activities (functions and tasks) according to functional and extra-functional considerations, by exploiting accurate operating condition profiles derived at design-time. However despite this object seems to be simple, run-time actions are based on three classes of information, namely: *functional status*, *extra-functional status* and *design-time configurations*.

Functional Status. The application's functional status is also referred as operating-mode. Given the requirement of the automotive application, a completely autonomous management system solely based on non-functional properties will not satisfy availability and functional needs. For this reason it is necessary to introduce the notion of operating mode, that expresses the current functional status of the system, e.g. the motion status of the vehicle or the status of the dashboard key. Associated to such states, different sets of functions shall be mandatorily enabled/disabled or properly configured, leaving to the non-functional manager the role of managing power (and other) optimizations, possibly at the cost of a processing quality degradation.

Extra-functional status. This information consists in the collection of metrics exposed by the extra-functional monitoring infrastructure. The framework provides to an appli-

cation the ability to monitor at runtime the desired metrics with a function-level granularity. The CONTREX monitoring framework is based on four main concepts. (i) Device. It is a physical component of the system that can be profiled in terms of extra-functional properties; (ii) Metric. It is any extra-functional property relevant for the application, such as time and energy, amount of data transferred by the system. (iii) Measure. It is defined by the metric, a numeric value and the related device. The metrics and the devices must be defined in the configuration of the framework. (iv) Event. It is meant to express concepts such as "the frequency of the CPU has changed to 200 MHz" or "the UART consumed 20 mJ". Additionally, in order to gather the measures, the developer must instrument the region of code that he wants to observe. In this way, the framework automatically senses the system and stores the observation values making them available to specific portions of the application responsible to implement local run-time management and to export them up to the cloud.

Design-time configurations. The design-time configuration depends on the results of application and node event-driven simulation, combined with user-defined policies explicitly specified by the application's developer. A system characterization framework of both hardware and software components have been developed and integrated in a sensor node-simulations. The characterization phase has been based on models, simulations and measurements on sub-systems and components.

C. Battery Modeling

Energy storage devices have a crucial role in determining the *lifetime of a system*, i.e., how long the system can operate autonomously from the grid or from power sources. This makes the modeling and simulation of energy systems, and of batteries in particular, an important dimension in system design. Monitoring or simulating the energy flows in the system would indeed allow an accurate estimation of energy consumption, and it would provide a forecast of system lifetime. Within the CONTREX project, battery models have been developed to be easily integrated inside an extra-functional monitoring framework. The adopted implementation language is SystemC, with its AMS extension, that can be easily integrated in C++ environments [26]. *Battery model* implementation strictly depends on the chosen level of detail [27]: *Functional models* implement the evolution of the energy flow through a function (e.g. equations or power state machines). *Circuit models* emulate the behavior of a battery through an equivalent electrical circuit (e.g., resistors and capacitors).

D. Automotive Demonstrator

Car Black Box (also called Event Data Recorder) is gaining an important role not only in investigation of car accidents or to track driver behavior, but also for a more direct user support. In fact, while it is true that it can be used to record the events and actions of the driver including speed, braking, turning, etc. seconds before a collision, thus possibly helping both the police and insurance companies in accident reconstruction, it

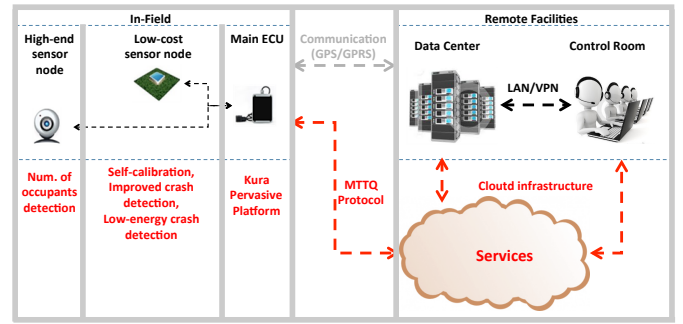


Fig. 8. Overview of the Automotive Demonstrator

can be also used to trigger an event when this negative events happens. In this direction, several non-automotive companies provide private and/or fleet vehicle drivers with a support service in case of accident [28]. This can happen calling directly either the mechanical support or an ambulance depending on the crash entity, or notifying the car owner if something happened to the car while it was in a parking area. From more technical point of view, the architecture is based on three main components: (i) a sensing unit for acceleration measurements, (ii) a data processing unit for GPS reading and accidents identification and (iii) a communication section for transmitting data to public authorities (hospital, police) or private support providers.

The Automotive Telematics Demonstrator of CONTREX extends this type of commercial solutions trying to provide more functionalities and better performances/power trade-off. In particular in the following part of the section we separate the description into functional and extra-functional enhancements.

1) *Functional improvements:* From the functional point of view the enhancement in the system architecture introduced by the CONTREX project are multiple (see Fig. 8).

First the car sensing unit has been enhanced by using new platform hardware and enhanced algorithms for event measurements. From the hardware point of view, the platform of the sensing unit have been migrated on the iNEMO-M1 [29], a 9-axis motion sensing System on-Board (SoB) guaranteeing high-definition digital acquisitions and ultra low power modes. On the software point of view the new capability introduced by the hardware platform have been exploited introducing new functionalities. In particular, a novel self-calibration algorithm has been defined to reduce the cost of installation. The device is now capable to auto-detect the positioning and orientation inside the car and autonomously auto-adapt the computation, thus not needing the intervention of highly qualified personnel neither for the installation nor for the maintenance. In addition, the new combination of the hardware and novel algorithms, enabled the possibility to detect low-energy events, such as minor crashes and acts of vandalism, while the car is parked and unattended.

Second, a high-end video sensing node - based on the ST SeCSoc ultra low-power computing platform - has been added to have visual information form inside the car, by detecting

the number of car occupants at the moment of an accident. The automatic detection and counting of vehicle occupants is a challenging problem within the Automotive demonstrator since it gives the possibility either to tailor the assistance in case of a crash based on the number and condition of vehicle occupants or to detect fraudulent behaviors.

Third, on the communication and data processing side, a new automotive gateway [30] has been included on-board to collect and transmit remotely the data gathered from the previously two described sensing nodes. This new device is a compact size device designed to support M2M applications and to host the Kura framework as described in Section V-A.

Fourth, a cloud infrastructure substitutes a custom datacenter to enable services scalability. In fact, being the number of customers and the amount of data to be collected per customer expected to significantly grow in the next few years, a switch from a dedicated server infrastructure to a flexible and scalable cloud-based solution is necessary, also considering the additional services highlighted in Section V-A.

2) *Extra-functional improvements*: In addition to the reduction of the installation cost given by the novel self-calibration algorithm mentioned in the previous section, other extra-functional improvements have been integrated in the automotive use case at the sensing-node level. In particular the introduction of the extra-functional property management framework described in Section V-B, while considering also the effect of the battery (Section V-C), enabled to improve the availability of the sensing devices even when the car is switched off.

VI. CONCLUSION

In this paper, we presented the preliminary results of the European project CONTREX. The developed tools and methodology extensions have been described as well as their application to the industrial demonstrators to show the benefits for the design of embedded mixed-criticality systems. The paper describes the technologies, languages, formalisms, and tools involved. More details about the project and its outcomes can be found at the project website [1].

Acknowledgements

This work has been supported by the EU integrated project CONTREX (FP7-611146).

REFERENCES

- [1] OFFIS, "CONTREX FP7 project website," 2015. [Online]. Available: <https://contrex.offis.de/home/>
- [2] I. Sander and A. Jantsch, "System modeling and transformational design refinement in ForSyDe," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, 2004.
- [3] "CAMEL-View Website," <http://www.ixtronics.com/21/index.html>, April 2016, last visited on 06/04/2016.
- [4] "D2.2.2: CONTREX system modelling methodology (final)," CONTREX Consortium, Tech. Rep., 2016. [Online]. Available: <https://contrex.offis.de/home/index.php/dissemination/deliverables>
- [5] H. Posadas, P. Peñil, A. Nicolás, and E. Villar, "Automatic synthesis of embedded SW for evaluating physical implementation alternatives from UML/MARTE models supporting memory space separation," *Microelectronics Journal*, vol. 45, no. 10, pp. 1281 – 1291, 2014.
- [6] F. Herrera, P. Peñil, and E. Villar, "A model-based, single-source approach to design-space exploration and synthesis of mixed-criticality systems," in *Proc. of the 18th Int. Workshop on Software and Compilers for Embedded Systems (SCOPES'15)*, New York, USA, 2015, pp. 88–91.
- [7] L. Diaz and P. Sanchez, "Host-compiled parallel simulation of many-core embedded systems," in *In Proc. of Design Automation Conference, DAC'14*, June 2014.
- [8] "VIPPE Website," <http://vippe.teisa.unican.es>, April 2016, last visited on 22/04/2016.
- [9] F. Herrera, P. Peñil, and E. Villar, "UML/MARTE modelling for design space exploration of mixed-criticality systems on top of predictable platforms," in *Jornadas Sarteco-JCE (JCE'15)*, September 2015.
- [10] P. Penil, H. Posadas, J. Medina, and E. Villar, "UML-based single-source approach for evaluation and optimization of mixed-critical embedded systems," in *DCIS'15*, November 2015.
- [11] "CONTREX system metamodel," CONTREX Consortium, Tech. Rep., 2015. [Online]. Available: <https://contrex.offis.de/home/images/publicdeliverables/Deliverable%20D2.1.1%20v1.0.pdf>
- [12] F. Herrera and E. Villar, "CONTREP: A single-source framework for UML-based modelling and design of mixed-criticality systems," in *DATE'16 University Booth*, September 2016.
- [13] E. A. Lee and A. Sangiovanni-Vincentelli, "A framework for comparing models of computation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, 1998.
- [14] S. Attarzadeh Niaki, M. Jakobsen, T. Sulonen, and I. Sander, "Formal heterogeneous system modeling with SystemC," in *Forum on Specification and Design Languages (FDL 2012)*, Vienna, Austria, 2012.
- [15] K. Rosvall and I. Sander, "A constraint-based design space exploration framework for real-time applications on MPSoCs," in *Design Automation and Test in Europe (DATE '14)*, Dresden, Germany, Mar. 2014.
- [16] F. Herrera and I. Sander, "An extensible infrastructure for modeling and time analysis of predictable embedded systems," in *FDL 2014*, Munich, Germany, Oct. 2014.
- [17] F. Castro, G. Palermo, C. Silvano, and V. Zaccaria, "Most: Multi-objective system tuner design space exploration for system architects," in *Proceedings of Designing for Embedded Parallel Computing Platforms: Architectures, Design Tools, and Applications*, March 2011.
- [18] "Open Virtual Platform." [Online]. Available: <http://www.ovpworld.org/>
- [19] K. Ebcioglu, E. Altman, M. Gschwind, and S. Sathaye, "Dynamic binary translation and optimization," *Computers, IEEE Transactions on*, vol. 50, no. 6, pp. 529–548, Jun 2001.
- [20] L. Cai and D. Gajski, "Transaction level modeling: An overview," in *Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES+ISSS '03. New York, NY, USA: ACM, 2003, pp. 19–24.
- [21] N. Bombieri, F. Fummi, V. Guarnieri, F. Stefanni, and S. Vinco, "Hdtlib: an efficient implementation of systemc data types for fast simulation at different abstraction levels," *Design Automation for Embedded Systems*, vol. 16, no. 2, pp. 115–135, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10617-012-9092-z>
- [22] N. Bombieri, F. Fummi, and G. Pravadelli, "Automatic Abstraction of RTL IPs into Equivalent TLM Descriptions," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1730–1743, Dec 2011.
- [23] A. Danese, G. Pravadelli, and I. Zandonà, "Automatic generation of power state machines through dynamic mining of temporal assertions," in *Design Automation and Test in Europe (DATE '16)*, Dresden, Germany, 2016.
- [24] P. A. Hartmann, K. Grüttner, and W. Nebel, "Advanced systemc tracing and analysis framework for extra-functional properties," in *The 11th International Symposium on Applied Reconfigurable Computing (ARC'15)*, 4 2015.
- [25] "Kura website," 2016. [Online]. Available: <http://www.eclipse.org/kura>
- [26] S. Vinco, A. Sassone, F. Fummi, E. Macii, and M. Poncino, "An open-source framework for formal specification and simulation of electrical energy systems," in *IEEE/ACM ISLPED*, 2014, pp. 287–290.
- [27] M. Petricca, D. Shin, A. Bocca, A. Macii, E. Macii, and M. Poncino, "An automated framework for generating variable-accuracy battery models from datasheet information," in *ACM/IEEE ISLPED*, 2013.
- [28] "Vodafone Automotive - Telematic Services," 2016. [Online]. Available: <http://www.cobra-group.com/vodafone-automotive>
- [29] "iNemo website," 2016. [Online]. Available: <http://www.st.com/inemo>
- [30] "EUTH IoT gateways website," 2016. [Online]. Available: <http://www.eurotech.com/en/products/devices/iot+gateways>